CS 13: Mathematical Foundations of Computing — Fall 2023

Course Syllabus

Information At-A-Glance

Instructor	
Name:	**Prof. Blank**
E-mail:	blank@caltech.edu
Office:	ANB 115
Office Hours:	Mon: 6pm - 8pm Tue: 5pm - 7pm
	Or by private meeting.

Course Website
https://math4cs.com
Visit early. Visit often.

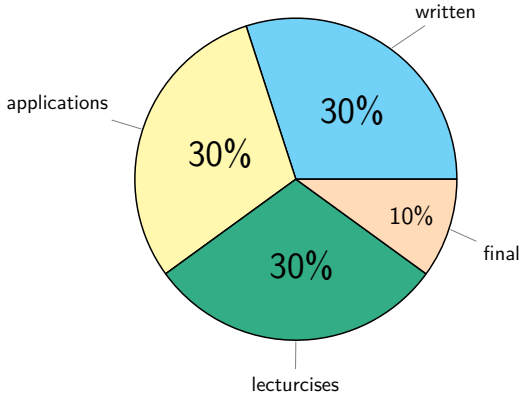
Lecture
ANB 105 on MWF
1:00 PM – 1:55 PM

Course Overview

This course is intended to prepare students for future work in proof-based CS courses like CS 21 and CS 38. It is organized around *cool computer science applications* of raw (mostly discrete) mathematical topics. CS 1 is a co-requisite as there will be a small number of programming assignments. We will cover basic set theory, induction and inductive structures (e.g., lists/trees), asymptotic analysis, combinatorics, number theory, and graph theory. Applications will include number representation, basic cryptography (RSA), basic algorithms on trees, numbers, and polynomials, graphs as social networks, compression and (simple) error-correcting codes.

Assessments

This course will consist of several types of assessments: written proof-based problem sets, verification problems, applications problem sets, and exams. We reserve the right to change these percentages and/or delete assessments at will.



Lecturcises

In most lectures, Prof. Blank will ask students to do "interactive exercises" some subset of which we will call out as "lecturcises". Every week, you will submit written solutions to at least *three* of the lecturcises for credit; each of these is worth approximately 1% of your final grade.

Since the lecturcises are during lecture, this means you'll have both an opportunity to collaborate on them AND hear a sketch of a solution from Prof. Blank! However, after lecture ends, **all lecturcises become completely nocollab with no TA help**. Note that, while you can collaborate during lecture, you should not **write your solutions during lecture**; instead, the write-up is expected to be solo. Though, you can obviously take notes about the solution. Note that if you do not attend a lecture, you can still turn in the lecturcises; they just become solo.

Grade Cutoffs

Please note that there is no way to receive a D in this course. Any score at or below 69% is considered an F, 69-70 is a D+, 70-80 is some kind of C, 80-90 is some kind of B, and 90-100 is some kind of A.

Drop Points

In lieu of a curve, we will give every student 70 "drop points", which means, in effect, we will add 70 points to your raw set score before computing final grades at the end of the quarter. Our goal with this policy is to provide leniency if, e.g., a student doesn't have time to complete a problem one week.

Set Minimums

In addition to raw score, to pass the course, you must get at least 20% of the points on all-but-one set. This means if you completely skip two or more sets in the course, you will not pass. Our goal here is to make sure students engage with *all* of the material without skipping topics as best as possible.

Written Sets

Unlike many of Prof. Blank's other courses, this one will have more "traditional" written problem sets. The goal here is help prepare you for solving more difficult computer science problems in future courses as well as improve your proof-writing skills.

Applications Sets

These sets will mix programming problems and proof-based written problems on the same topic to demonstrate how the proofs are useful to the application.

Verification Sets

More on these later.

Exams

The final exam will be **cumulative** with an equal emphasis on all the material in the course. They will be entirely written, proof-based questions. The time limit will be *13 hours* though we expect nobody will use anywhere near that much time.

Late Policy

In this course, you will earn one "late token" per week. On every Saturday of the term, your late token count will increment automatically. Each late token will allow you to submit a project up to 24 hours late; tokens are indivisible and you may not go into "token debt". You may not use more than two late tokens per project.

You do not need to use tokens for serious medical (physical or mental) or emotional circumstances; in such situations, contact the instructor to work out a plan for completing the work in a reasonable time frame.

Getting Help

Please don't be afraid to ask for help if you don't understand something. Prof. Blank holds *at least three* office hours a week, and they get lonely and bored if you don't show up! They also show up early to lecture and are happy to answer any questions you might have before or after lecture. At office hours, you can ask for clarification on a lecture (or for a *repetition* of the lecture!). You can ask for help with a frustrating part of the homework. You can even show up just to tell us you're frustrated and vent. Some first steps on how to get help are: ask on Ed, ask someone on course staff questions before/after lecture, and go to office hours.

Collaboration & Academic Integrity

See our "collaboration table" on the website. We reserve the right to modify or clarify this policy as needed. Notably, you may not, under any circumstances, look at another student's/group's proofs.